(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 199 44 991 A 1**

(51) Int. Cl.[7]: G 06 F 12/14

(21) Aktenzeichen: 199 44 991.0
(22) Anmeldetag: 20. 9. 1999
(43) Offenlegungstag: 12. 4. 2001

DE 199 44 991 A 1

(71) Anmelder:
Giesecke & Devrient GmbH, 81677 München, DE

(72) Erfinder:
Baldischweiler, Michael, 81825 München, DE

(56) Entgegenhaltungen:
DE 35 02 387 C2
DE 197 01 166 A1
US-Z: IEEE Transaction on Computer-Aided Design, Vol. 9, 6/90, S. 665-669;

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Verfahren zur Sicherung eines Programmablaufs

(57) Die Erfindung betrifft ein Verfahren zur Sicherung des Programmablaufs beim Aufruf von Unterprogrammen. Bekannte Verfahren zur Datensicherung wirken zwar einer Auswertung der Daten durch gezielte Programmunterbrechung entgegen, bieten jedoch keinen wirksamen Schutz bei modular aufgebauten Programmen, insbesondere beim Aufruf von Unterprogrammen. Gemäß der Erfindung führt deshalb das aufgerufene Programm vor bzw. während der Programmausführung eine Überprüfung der vom aufrufenden Programm direkt oder indirekt übermittelten Daten aus.

DE 199 44 991 A 1

Beschreibung

Die vorliegende Erfindung betrifft ein Verfahren zur Sicherung des Programmablaufs gemäß Anspruch 1.

Insbesondere bei sicherheitsrelevanten Anwendungen, beispielsweise im Bereich von IC-Karten ist es notwendig, den Programmablauf vor unerlaubten Manipulationen zu schützen. Zum Schutz geheimer Daten, beispielsweise geheimer Schlüsseldaten, ist es bekannt, die zu schützenden Daten verschlüsselt abzulegen, um ein Auslesen durch Unberechtigte zu verhindern.

Der Zugriff auf geheime Daten kann jedoch auch dadurch erfolgen, daß der Programmablauf gezielt unterbrochen wird, so daß Fehler in den Verschlüsselungsroutinen entstehen, aus denen nach mehrmaliger gezielter Unterbrechung auf die geheimen Daten rückgeschlossen werden kann.

Zur Vermeidung derartiger Angriffe ist es notwendig, Fehler oder Störungen des Programmablaufs sicher zu erkennen. Aus der deutschen Patentschrift DE 37 09 524 C2 ist ein Verfahren zur Überprüfung der Speicherzelleninhalte eines Programmspeichers in einem Rechner bekannt. Dort werden mehrere Prüfsummen abgespeichert, welche aus Speicherzelleninhalten von unterschiedlichen Adreß- und Datenspeicherbereichen gebildet werden. Die Prüfsummen werden zu Beginn und/oder während des Rechnerbetriebs ermittelt und mit der abgespeicherten Prüfsumme verglichen. Bei Feststellung einer Abweichung wird ein Fehlersignal ausgegeben.

Das aus der DE 37 09 524 C2 bekannte Verfahren eignet sich hauptsächlich dazu, die Richtigkeit von Daten, welche in einem Programm verwendet werden, zu überprüfen. Es wird außer Acht gelassen, daß auch bzw. insbesondere bei Programmaufrufen, d. h. bei der Ausführung von Unter- oder Funktionsprogrammen eine Manipulation des Programmablaufs erfolgen kann.

Es ist deshalb Aufgabe der vorliegenden Erfindung, ein Verfahren anzugeben, das die sichere Überprüfung von modular aufgebauten Programmen, insbesondere bei Unterprogrammaufrufen, zuläßt.

Gemäß der Erfindung wird diese Aufgabe gelöst, indem vom aufgerufenen Programm eine Datenüberprüfung ausgeführt wird, welche den sicheren Übergang der vom aufrufenden Programm zu übergebenden Daten feststellt.

Durch die Erfindung wird eine zusätzliche Sicherheit erreicht, die nicht nur gewährleistet, daß einzelne Programmteile sicher und vollständig ausgeführt werden, sondern daß der gesamte Programmablauf ungestört und frei von Manipulationen ablaufen kann.

Eine vorteilhafte Ausführungsform der Erfindung sieht vor, daß über die vom aufrufenden Programm an das aufgerufene Programm übergebenen Parameter zunächst vom aufrufenden Programm eine Checksumme gebildet wird, welche in einem dafür vorgesehenen Speicherbereich abgelegt wird. Nach Übergabe der Parameter wird auch vom aufgerufenen Programm über die erhaltenen Parameter eine Checksumme gebildet. Für den Fall, daß die vom aufrufenden und aufgerufenen Programm gebildeten Checksummen unterschiedlich sind, wird das Programm abgebrochen.

Auf diese Weise kann sichergestellt werden, daß ein Funktionsprogramm, insbesondere ein Funktionsprogramm, welches sicherheitsrelevante Daten abarbeitet, bereits zu Beginn auf Manipulationen hin untersucht wird, so daß der Start des aufgerufenen Programms mit fehlerhaften Parametern von vornherein verhindert werden kann und eine Auswertung der fehlerbehafteten Daten nicht ermöglicht wird.

Vorzugsweise wird der Speicherbereich, welcher zur Ablage der Checksumme vorgesehen ist, in einem RAM- oder Registerbereich angelegt.

Eine weitere oder alternative Ausführungsform zur Bildung der Checksumme über die zu übergebenden Parameter ergibt sich aus der Überprüfung der Rücksprungadressen. Dabei werden die Rücksprungadressen der aufrufenden Funktionen in einer Tabelle eingetragen und vom aufgerufenen Programm kann mittels dieser Tabelle überprüft werden, ob die vom aufrufenden Programm übermittelte Rücksprungadresse in der Tabelle vorhanden ist. Bei einer fehlerhaft mitgeteilten Rücksprungadresse kann das Programm unterbrochen werden.

Eine weitere alternative oder zusätzliche Sicherheitsüberprüfung kann erfolgen, indem bei Aufruf eines Unterprogramms bzw. eines Funktionsprogramms ein Timer gestartet wird. Dieser Timer zählt die Taktzyklen, welche für die Ausführung des Programms notwendig sind. Es wird dabei zunächst als Grenzwert für den Timer die für den regulären Unterprogrammablauf benötigte Anzahl der Taktzyklen als Grenzwert vorgegeben. Das Programm wird abgebrochen, wenn vor Beendigung des Unterprogramms die Anzahl der vorgegebenen Taktzyklen überschritten wurde.

In vorteilhafter Weise wird auch an bestimmten, vorgegebenen Stellen des Unterprogramms der Timerwert ausgelesen und mit ebenfalls vorgegebenen Zwischenwerten verglichen. Auch in diesem Fall wird das Programm abgebrochen, wenn der vorgegebene Zwischenwert überschritten wurde.

Im folgenden wird die Erfindung anhand der **Fig.** 1 bis 3 näher erläutert.

Es zeigen:

Fig. 1 Ablaufdiagramm für die Überprüfung mittels Prüfsumme,

Fig. 2 Ablauf für die Überprüfung mittels Rücksprungadressentabelle,

Fig. 3 Ablauf für die Überprüfung mittels Timer.

In **Fig.** 1 ist der Ablauf eines Unterprogrammaufrufs, insbesondere eines Funktionsaufrufs beschrieben, wobei die Funktionsschritte **1** bis **3** das aufzurufende Programm betreffen und die Funktionsschritte **4** bis **8** die Auswertung des Unterprogramms betreffen.

Im aufzurufenden Programm werden zunächst in Schritt **1** die für die Ausführung des Unterprogramms notwendigen Parameter bereitgestellt. Für diese Parameter wird in Schritt **2** eine Prüfsumme gebildet, die im einfachsten Fall aus einem Parity-check bestehen kann. Im weiteren sind selbstverständlich die gängigen Verfahren zur Prüfsummenbildung, z. B. CRC (Cyclical Redundancy Check) oder EDC einsetzbar. Die so ermittelte Prüfsumme (Checksumme) wird in einen dafür vorgesehenen Speicherbereich eingeschrieben. Bei diesem Speicherbereich kann es sich um einen flüchtigen Speicher (RAM) oder auch um einen nichtflüchtigen, wiederbeschreibbaren Speicher (z. B. EEPROM) handeln.

In Anschluß an die Bildung und Abspeicherung der Prüfsumme **1** erfolgt der Unterprogrammaufruf in Schritt **3**. Schritt **4** stellt den Beginn der Ausführung des Unterprogramms dar. In diesem Unterprogramm wird zunächst die Prüfsumme **2** über die übergebenen Parameter gebildet. Diese Prüfsumme wird mit dem gleichen Verfahren gebildet, mit dem auch die Prüfsumme **1** im aufrufenden Programm ermittelt wurde.

Als nächstes erfolgt in Schritt **6** eine Überprüfung der Prüfsummen PS1 und PS2 auf Gleichheit. Wird in diesem Schritt **6** festgestellt, daß die beiden Prüfsummen ungleich sind, kann davon ausgegangen werden, daß bei der Übergabe der Programmparameter ein Fehler aufgetreten ist, welcher ein Hinweis auf eine beabsichtigte Störung mit dem Ziel, Geheimdaten zu ermitteln, sein kann. Als Maßnahme kann in Schritt **7** das Programm beendet werden oder es werden entsprechende alternative Maßnahmen getroffen,

beispielsweise eine Fehlermeldung an das Hauptprogramm.

Wird in Schritt **6** festgestellt, daß die Prüfsummen PS1 und PS2 gleich sind, wird mit der eigentlichen Funktionsausführung begonnen.

Die **Fig.** 2 zeigt eine Möglichkeit der Programmsicherung durch Überprüfen der Rücksprungadressen. Rücksprungadressen werden beim Funktionsaufruf per Hardware auf den Stack gelegt. Im vorliegenden Fall werden also im Schritt **11** beim Unterprogrammaufruf, ebenfalls die Informationen vom aufrufenden Programm (z. B. Rücksprungadressen) an das Unterprogramm übergeben. Gemäß der Erfindung werden die Rücksprungadressen in einer Tabelle 17 verwaltet und bei der Ausführung des Unterprogramms werden zunächst in Schritt **12** die Rücksprungadressen – soweit sie im RAM abgelegt sind – auf Konsistenz hin untersucht, um sie in Schritt **13** anhand der Tabelle 17 zu überprüfen. Wenn in Schritt **14** festgestellt wurde, daß die übergebene Rücksprungadresse nicht in der Tabelle vorhanden ist, wird mit Schritt **15** das Programm beendet, andernfalls wird in Schritt **16** mit der Ausführung des Funktionsprogramms begonnen.

Die **Fig.** 3 zeigt eine Ausführungsform, bei der der richtige Programmablauf bzw. der ungestörte Programmablauf mittels eines Timers überprüft wird. Unmittelbar nach dem Start des Unterprogramms in Schritt **21** wird in Schritt **22** ein Timer gestartet. Dieser Timer ist ausgelegt, die Zeit zu messen bzw. die Taktzyklen zu zählen, welche für die Ausführung des Unterprogramms benötigt werden. Im Anschluß an den Start des Timers in Schritt **22** wird mit Schritt **23** die Funktion des Unterprogramms ausgeführt und nach Beendigung der Funktion wird in Schritt **24** der Timer gestoppt. In Schritt **25** wird überprüft, ob die Anzahl der Taktzyklen, welche für die Ausführung des Funktionsprogramms benötigt wurden, mit der vorgegebenen Anzahl von Taktzyklen übereinstimmt. Für den Fall, daß keine Übereinstimmung besteht, wird das Programm mit Schritt **26** beendet. Im anderen Fall wird in Schritt **27** mit der Programmausführung fortgefahren, beispielsweise indem zum Hauptprogramm zurückgesprungen wird.

In der **Fig.** 3 ist dargestellt, daß der Timer nach Ablauf der Funktion bzw. des Funktionsprogramms gestoppt und überprüft wird. In der Praxis kann die Sicherheit erhöht werden, indem bestimmte Stellen im Funktionsprogramm vorgesehen werden, an denen der Timer zusätzlich überprüft wird. Damit kann gegebenenfalls verhindert werden, daß das Funktionsprogramm trotz eines Fehlers oder Angriffs weitgehend ausgeführt wird.

Alternativ kann auch vorgesehen werden, daß der Timerwert nach dem Start kontinuierlich mit einem Grenzwert verglichen wird und das Programm abgebrochen wird, wenn dieser Grenzwert erreicht bzw. überschritten wurde.

Die einzelnen Ausführungsbeispiele nach den **Fig.** 1 bis 3 wurden als eigenständige, alternative Maßnahmen dargestellt. Die Sicherheit kann erhöht werden, indem die Ausführungsbeispiele kombiniert werden. Größte Sicherheit bildet die parallele Überprüfung mittels Prüfsumme, Rücksprungadressenprüfung und Timerüberprüfung.

Patentansprüche

1. Verfahren zur Sicherung des Programmablaufs beim Aufruf von Unterprogrammen, **dadurch gekennzeichnet**, daß das aufgerufene Programm vor bzw. während der Programmausführung eine Überprüfung der vom aufrufenden Programm direkt oder indirekt übermittelten Daten ausführt.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß
   - das aufrufende Programm über die zu übergebenden Parameter eine erste Checksumme bildet,
   - diese erste Checksumme in einem dafür vorgesehenen Speicherbereich abgelegt wird,
   - das aufgerufene Programm vor seiner Ausführung über die erhaltenen Parameter eine zweite Checksumme bildet und auf Gleichheit mit der ersten Checksumme überprüft und
   - bei Ungleichheit der ersten und der zweiten Checksumme das Programm abgebrochen oder eine Fehlermeldung ausgegeben wird.

3. Verfahren nach Anspruch 1 oder Anspruch 2, dadurch gekennzeichnet, daß der Speicherbereich zur Ablage der Checksumme ein RAM- oder Registerbereich ist.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß die Rücksprungadressen der aufrufenden Funktion in einer Tabelle eingetragen werden und das aufgerufene Programm die vom aufrufenden Programm mitgeteilte Rücksprungadresse überprüft, indem das Vorhandensein dieser Rücksprungadresse anhand der Tabelle überprüft wird.

5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß bei Aufruf eines Unterprogramms ein Timer gestartet wird, welcher die für die Ausführung des Programms benötigte Anzahl von Taktzyklen zählt und das Programm abbricht, wenn vor Beendigung des Unterprogramms die vorgegebene Anzahl der Taktzyklen überschritten wurde.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß der Timerwert an bestimmten, vorgegebenen Stellen ausgelesen und mit einem ebenfalls vorgegebenen Zwischenwert verglichen wird und das Programm abgebrochen wird, wenn der vorgegebene Zwischenwert überschritten wurde.

Hierzu 2 Seite(n) Zeichnungen

- Leerseite -

Int. Cl.[7]: G 06 F 12/14
Offenlegungstag: 12. April 2001

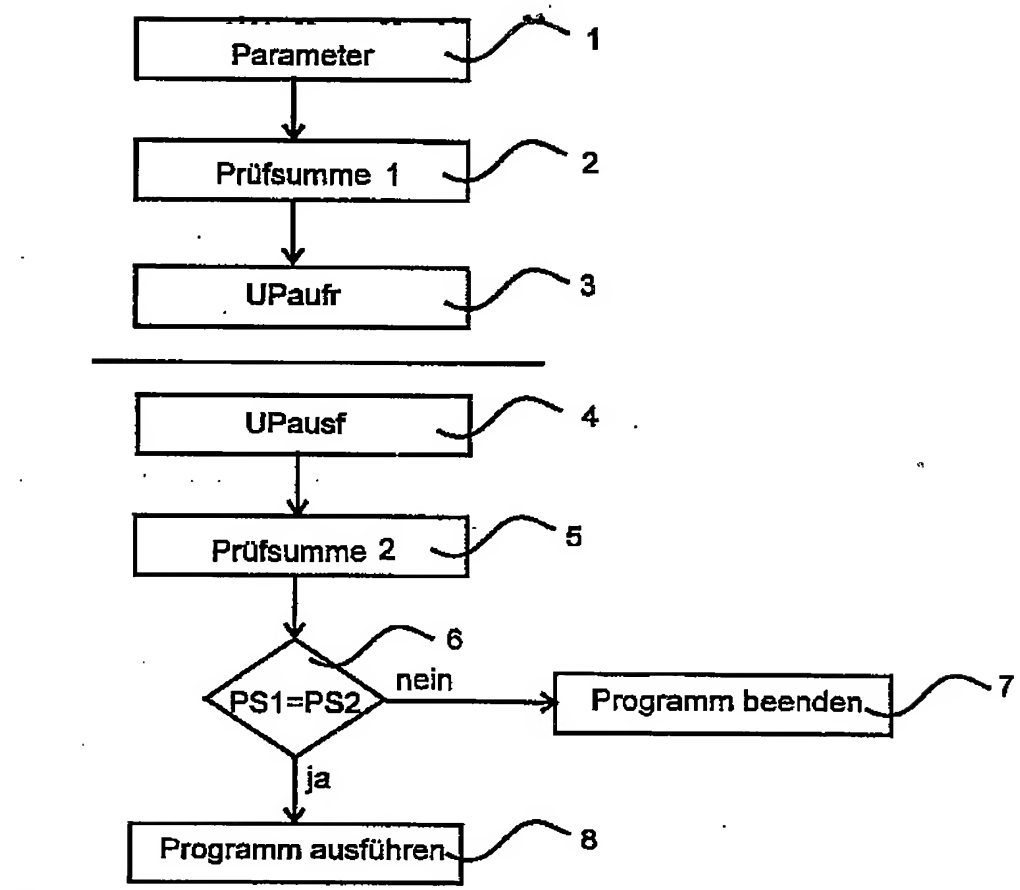

Fig. 1

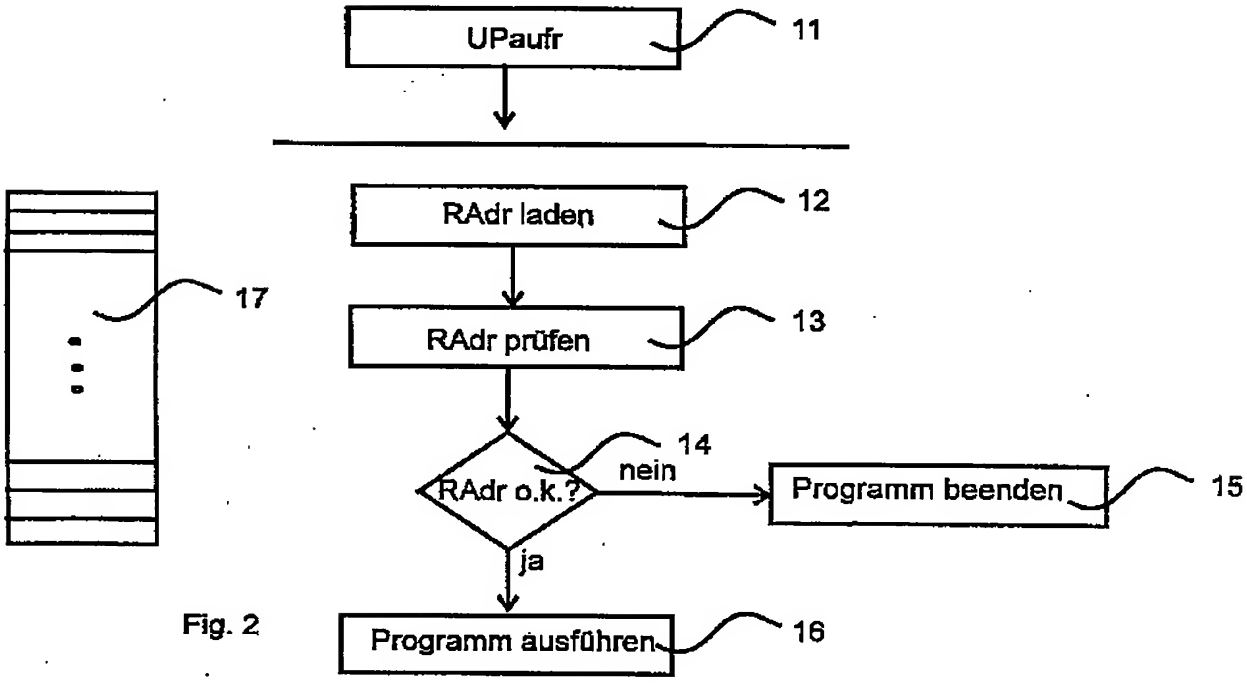

Fig. 2

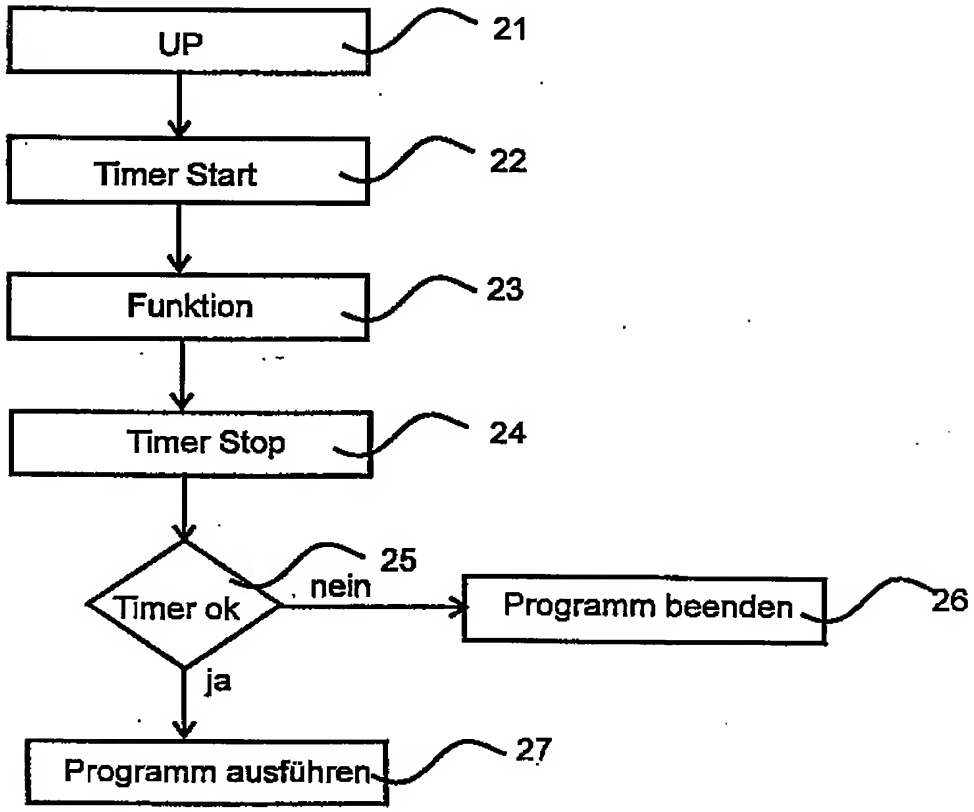
UP
21
Timer Start
22
Funktion
23
Timer Stop
24
Timer ok
25
nein
Programm beenden
26
ja
Programm ausführen
27

Fig. 3